

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 58-115569

(43)Date of publication of application : 09.07.1983

(51)Int.Cl.

G06F 15/16

(21)Application number : 56-212082

(71)Applicant : FUJI ELECTRIC CO LTD
FUJI FACOM CORP

(22)Date of filing : 29.12.1981

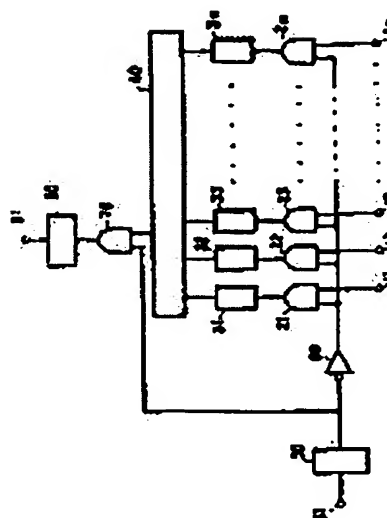
(72)Inventor : HAGIWARA KENICHI

(54) MULTIPROCESSOR SYSTEM

(57)Abstract:

PURPOSE: To improve the throughput of the whole of the system, by recognizing immediately the existence of an idle processor by a main processor and assigning this processor to a task to which the start request is issued.

CONSTITUTION: Contents of flag registers of plural processors constituting the multiprocessor system are inputted to input terminals 11,12...1n corresponding to idle processors. These terminals are connected to one terminals of corresponding gate circuits 21,22...2n, and the other input terminals are connected to a processor select indication flag register 50 through an inverter 60. Meanwhile, output terminals of circuits 21W2n are connected to input terminals of corresponding processor status display flag registers 31,32...3n, and outputs of these registers are connected to a processor number register 80 through a priority encoder 40 and a gate circuit 70. Thus, the virtual device having the channel bus is provided to control exclusively the specific area to the mass storage device by the channel bus.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

⑨ 日本国特許庁 (JP)
⑩ 公開特許公報 (A)

⑪ 特許出願公開
昭58—115569

⑫ Int. Cl.³
G 06 F 15/16

識別記号

庁内整理番号
6619—5B

⑬ 公開 昭和58年(1983)7月9日

発明の数 1
審査請求 未請求

(全 6 頁)

⑭ マルチプロセッサ方式

⑮ 特 願 昭56—212082

⑯ 出 願 昭56(1981)12月29日

⑰ 発 明 者 萩原賢一

日野市富士町1番地富士ファコ

△ 制御株式会社内

⑱ 出 願 人 富士電機製造株式会社

川崎市川崎区田辺新田1番1号

⑲ 出 願 人 富士ファコム制御株式会社

日野市富士町1番地

⑳ 代 理 人 弁理士 玉蟲久五郎 外3名

明 細 書

1 発明の名称

マルチプロセッサ方式

2 特許請求の範囲

マスタプロセッサ及びこれに従属する複数のプロセッサから構成されるマルチプロセッサ・システムにおいて、

各プロセッサがタスクを終了したときに該プロセッサが空き状態になったことを表示するプロセッサ状態表示フラグを該プロセッサから受けかつ各プロセッサがタスクに割付けられたときに該プロセッサがタスク実行状態になったことを表示するプロセッサ状態表示フラグを該プロセッサ又はマスタプロセッサのCPUから受け、該受けたプロセッサ状態表示フラグをマスタプロセッサのCPUの制御に基いて格納するプロセッサ対応に設けられたレジスタ、該プロセッサ対応に設けられたレジスタの内容に基き空き状態にあるプロセッサが存在する場合には所定の選択基準に従って空き状態にあるいずれか1台のプロセッサを選択して該

プロセッサ番号を出力しかつ空き状態にあるプロセッサが存在しない場合にはその旨を表示する信号を出力する選択手段、及びタスク管理テーブルをマスタプロセッサに備え、

マスタプロセッサは、各プロセッサからタスクの起動要求を受けたときは前記選択手段の出力信号を検出し、該出力信号がプロセッサ番号を表示している場合には該番号を有するプロセッサを前記起動要求に係るタスクに割付け、前記出力信号が空きプロセッサの不存在を表示している場合には前記タスク管理テーブルに従って選択したプロセッサを前記起動要求に係るタスクに割付けることを特徴とするマルチプロセッサ方式。

3 発明の詳細な説明

この発明はプロセッサ割付け方式の改良により処理能力の向上を図ったマルチプロセッサ方式に関するものである。

複数台のプロセッサから構成されるマルチプロセッサ・システムにおいては、タスク(特定の目的をもった処理の集合)の割付けに関するプロセ

ツナ相互間の競合解決手段として、マスタプロセス方式が採用されている。この方式は、複数台のプロセッサのうちいずれか1台をマスタプロセスとし、これに従属する残りの各プロセッサは、他のプロセッサへの割付けを伴うタスクの実行要求が生じたときにはその旨をマスタプロセスに通知し、この通知を受けたマスタプロセスがそのタスクの優先度、システム稼働状態などに応じこの実行要求タスクに対して最適のプロセッサを割付ける構成となつている。

従来この種マスタプロセス方式におけるプロセス割付け処理は、すべてのタスクの状態を管理するためのタスク管理テーブル及びすべてのプロセッサの状態を管理するためのプロセッサ管理テーブルをマスタプロセス内に備え、各プロセッサからのタスク実行要求に対し、マスタプロセスが以下に示すような手順に従つて割付けるべきプロセッサを選択するというソフトウェアによる方式が採用されている。すなわち、あるタスクに対する起動要求が通知されると、マスタプロセ

ッサはプロセッサ管理テーブルを調べて、タスクを実行していない空き状態のプロセッサを探す。空き状態のプロセッサが見つければそのプロセッサを起動要求のあつたタスクに割付ける。一方、空き状態のプロセッサが見つからなければ、タスク管理テーブルを調べて起動要求のあつたタスクよりも優先順位の低いタスクを実行しているプロセッサを探す。もし、そのようなプロセッサが見つければそのプロセッサが実行中のタスクを中断させ、起動要求のあつた高い優先順位のタスクを実行させる。一方、そのような低優先順位のタスクを実行しているプロセッサが見つからなければ、起動要求は保留される。上記プロセス割付け作業の結果は、タスク管理テーブル及びプロセッサ管理テーブルに記録される。特に、新たにタスクが割付けられたプロセッサについては、プロセッサ管理テーブルに「タスク実行状態」が記録される。

また各プロセッサは実行していたタスクを終了するたびに、その旨をマスタプロセスに通知す

る。この通知を受けたマスタプロセスのCPUは、タスク管理テーブル及びプロセッサ管理テーブルにその旨を記録する。特にタスクを終了したプロセッサについては、プロセッサ管理テーブルに「空き状態」が記録される。その後マスタプロセスのCPUはタスク管理テーブルを調べ、起動要求が保留中のタスクあるいは優先順位が低かつたため実行が中断されたタスクが存在していれば、そのうちいずれかを選択し、選択したタスクにプロセッサを割付ける作業を行なう。

しかしながら、上記マスタプロセス方式におけるプロセス割付け方式には次のような欠点がある。すなわち、タスク起動要求の通知及びタスク終了の通知はマスタプロセスのCPUで到着順に1つずつ処理されるので、起動要求のあつたタスクにプロセッサを割付ける作業中に、タスクの終了に伴つて空き状態になつたプロセッサが発生した場合であつてもその旨の通知がマスタプロセスで保留されることになるので、マスタプロセスは新たに発生した空き状態を直ちに認識する

ことができない。このためマスタプロセスは、空き状態のプロセッサが実際に存在するにも拘らず、優先順位の低いタスクを実行しているプロセッサをあえて中断させ、このプロセッサに実行要求のあつたプロセッサを割付けてしまうという不合理を生ずる場合がある。

第1図はこのような一例を示す概念図であり、横軸は時間を示し、縦軸はプロセッサ及びこれらが実行中のタスクの区分を示している。第1図において、7台のプロセッサ $P_1 \sim P_7$ のうち P_1 がマスタプロセスであり、これに従属する各プロセッサ $P_2 \sim P_7$ は時刻 t_0 においてそれぞれタスク $T_2 \sim T_7$ を実行している。これらタスクは、 $T_2, T_3, T_4, T_5, T_6, T_7$ の順に優先順位が低くなるものとする。時刻 t_1 においてプロセッサ P_2 が実行中のタスク T_2 が新たなタスク T_8 の起動要求を発生する。これを受付けたマスタプロセス P_1 は上述した手順に従つてプロセス割付け作業を開始する。この割付け作業中に時刻 t_2 においてプロセッサ P_2 の実行中のタスク T_2 が新たなタスク T_9 の起動要求を発生する

が、これはマスタプロセッサ P_1 において保留される。引続いて時刻 t_2 においてプロセッサ P_2 が実行していたタスク T_2 が終了してタスク終了通知が発せられるが、これもマスタプロセッサ P_1 において保留される。マスタプロセッサ P_1 は、時刻 t_4 において、優先順位の最も低いタスク T_1 を実行しているプロセッサ P_1 を選択し、これをプロセッサ P_2 から起動要求のあつたタスク T_2 に割付ける。ただしタスク T_1 に比べてタスク T_2 の優先順位が高いものとする。

マスタプロセッサの CPU は、タスク T_2 に対するプロセッサ P_2 の割付けが終了すると、保留しておいたプロセッサ P_2 からの起動要求の処理を開始する。このプロセッサ割付け処理は上述した手順と全く同様の手順で行われる。この場合、プロセッサ P_2 からのタスク T_2 の終了通知が依然として保留されたままになっているので、マスタプロセッサはプロセッサ P_2 が実際には空き状態にあることを認識できない。このため、マスタプロセッサは、時刻 t_4 において、最も優先順位の低いタスク T_1 を

実行しているプロセッサ P_2 のタスクを中断させてこれをタスク T_1 に割付け、この割付けが終了した後、はじめてプロセッサ P_2 からのタスク終了通知の処理を開始することになる。

このように、従来方式においてはプロセッサからのタスク終了通知をタスク起動要求と同様に到着順にマスタプロセッサで処理していたので、マスタプロセッサが空き状態になつたプロセッサの存在を認識するまで長時間を要することもあり、システム全体の処理能力を十分に発揮できないという欠点があつた。

本発明は上記従来方式の欠点を考慮してなされたものであり、その目的は、マスタプロセッサが空き状態になつたプロセッサの存在を直ちに認識して、これを起動要求のあつたタスクに割付けることによつてシステム全体の処理能力の向上を図つたプロセッサ割付け方式を提供することにある。

以下本発明の詳細を実施例により説明する。

第2図は、本発明の一実施例に使用する空きプロセッサ選択装置の構成の一例を示すブロック図

であり、これはマスタプロセッサ内に設けられている。マルチプロセッサを構成する n 台のプロセッサは、それぞれタスク実行中であるか空き状態にあるかを表す状態フラグ・レジスタ（図示せず）を備えており、これらのレジスタの内容はそれぞれこの空きプロセッサ選択装置の対応の入力端子 11、12、13...1 n に入力する。これら入力端子はそれぞれ対応のゲート回路 21、22、23...2 n の一方の入力端子に結合されており、これらゲート回路の他方の入力端子はインバータ 60 を介してプロセッサ選択指示フラグ・レジスタ 50 の出力端子に結合されている。ゲート回路 21、22、23...2 n の出力端子は、対応のプロセッサ状態表示フラグ・レジスタ 31、32、33...3 n の入力端子に結合され、これらレジスタの出力端子はプライオリティ・エンコーダ 40 の入力端子に結合されている。このプライオリティ・エンコーダの出力端子はゲート回路 70 の一方の入力端子に結合されており、このゲート回路の他方の入力端子は、プロセッサ選択指示フラグ・レジスタ 50

の出力端子に直結されている。ゲート回路 70 の出力端子はプロセッサ番号レジスタ 80 の入力端子に結合されており、このレジスタ 80 の内容は端子 81 を介してマスタプロセッサの CPU に送出される。

入力端子 11 (1=1~ n) には、対応のプロセッサ i がタスクに割付けられたときにそのプロセッサがタスク実行状態になつたことを示すフラグ「0」がマスタプロセッサの CPU から供給され、一方対応のプロセッサ i がタスクを終了したときにそのプロセッサが空き状態になつたことを示すフラグ「1」がマスタプロセッサの CPU を介することなくプロセッサ i から直接供給される。マスタプロセッサは、上述のように構成された空きプロセッサ選択装置の他に、すべてのタスクの進行状態を管理するためのタスク管理テーブル（図示せず）を備えている。

マスタプロセッサは、いずれかのプロセッサからタスクに対する起動要求の通知を受けると、まずプロセッサ選択指令フラグ・レジスタ 50 にブ

プロセスの選択を指令するフラグ「1」をセットする。これによつてすべてのゲート回路21~2nが遮断され、すべてのプロセス状態表示フラグ・レジスタ31~3nが対応の入力端子11~1nから分離されてその内容の更新が禁止される。プライオリティ・エンコーダ40は、プロセス状態表示フラグ・レジスタ31~3nからプロセス状態表示フラグを読み込み、プロセスの空き状態を表示するフラグ「1」が1個だけ存在する場合にはそのフラグを出力しているレジスタ31の収容位置情報（プロセス番号）を出力し、プロセスの空き状態を表示するフラグ「1」が複数個存在する場合にはそれらの中から所定の優先順位に従つて1個を選択しそのプロセス番号を出力する。一方プライオリティ・エンコーダ40は、送込んだプロセス状態表示フラグがすべてタスク実行中を表示する「0」を出力している場合には、「0」を出力する。プライオリティ・エンコーダ40から出力された空き状態にあるプロセス番号又はすべてのプロセスがタスク実行中であることを表

示する「0」は、ゲート70を介してプロセス番号レジスタ80にセットされる。

マスタプロセスのCPUは、更新されたプロセス番号レジスタ80の内容を出力端子81を介して読取り、これが「0」でなく空き状態にあるプロセスの番号を表示している場合には、その番号に対応するプロセスを起動要求タスクに割付ける。一方マスタプロセスのCPUは、プロセス番号レジスタ80の内容が「0」であつて空きプロセスが存在しない場合には、タスク管理テーブルの内容に基づいて、起動要求のあつたタスクに比べて低い優先順位のタスクを実行しているプロセスをさがす。そのようなプロセスが存在すればその実行中のタスクが中断されてこれに起動要求タスクが割付けられ、そのようなプロセスが存在しなければ起動要求は保留される。マスタプロセスのCPUは、空き状態にあつたプロセスに起動要求のあつたタスクを割付けた場合には、このプロセスに対応する入力端子11にタスクの実行状態を示すフラグ「0」を出力し、引続いて

プロセス選択指令フラグ・レジスタ50に「0」をセットする。これにより、入力端子11~1n上の新たなプロセス状態表示フラグがゲート21~2nを介してレジスタ31~3nにセットされる。

一方各プロセスは、実行していたタスクを終了したときには、まず対応の入力端子11上にプロセスの空き状態を表示するフラグ「1」を出力し、引続いてタスクの終了をマスタプロセスのCPUに通知する。入力端子11上に出力されたフラグ「1」は、CPUがプロセス選択中でなければ直ちに、選択中であればこれが終了するのをまつて、ゲート21を介してレジスタ31にセットされる。一方マスタプロセスのCPUに通知されたタスク終了通知は、他のプロセスからの終了通知あるいは起動要求などの到着順序に従つて直ちにあるいは適宜な時間保留されたのち処理され、タスク管理テーブルの内容が更新される。引続いて、マスタプロセスのCPUはタスク管理テーブルを調べ、起動要求が保留されているタスクあるいは優先順位が低いため中断されていたタスクが

存在すれば、そのうちのいずれか1つを通宜な選択基準に従つて選択し、これに空き状態になつたプロセスを割付けるための作業を行う。

第3図は、本発明の一実施例に使用する空きプロセス選択装置の他の構成の一例を示すブロック図である。この装置は第2図示の装置と同様マスタプロセス内に設けられている。第3図において、第2図と同一の参照符号を付した要素は、第2図に関し既に説明したものと同一であるから、これらについては重複した説明を要しないであろう。第3図中91、92...9nは対応のプロセス1、2...nの動作可能表示フラグをマスタプロセスのCPUから受ける入力端子、101、102...10nはこれら入力端子のフラグを格納する動作可能表示フラグ・レジスタ、111、112...11nはゲート回路である。

マスタプロセスのCPUは、起動要求のあつたタスクにプロセスを割付けるための前述した作業において、プロセス選択指令フラグ・レジスタ50に「1」をセットするに先立つて、タスク管

理テーブルに保持されているタスクごとに定められた動作可能表示フラグを読出し、これを入力端子 91~9n を介して動作可能表示フラグ・レジスタ 101~10n にセットする。この動作可能表示フラグは、現在要求されているタスクに対し動作が許容されているプロセッサについては「1」が割当てられており、その他のプロセッサについては「0」が割当てられている。マスタプロセッサの CPU は、動作可能表示フラグをレジスタ 101~10n にセットしたのち、入力端子 51 を介してプロセッサ選択指示フラグ・レジスタ 50 に「1」をセットする。以後、第 2 図に同じ説明したと同様の動作が行われ、実行要求があつたタスクに割付けるべきプロセッサが選択される。従つてこの構成においては、空き状態にありかつタスクに対する動作が許容されているプロセッサを選択することができるので、空き状態にあるプロセッサを一旦選択したのちこれについてタスクに対する動作の可否を判断する手順が不要になり、選択時間の短縮が可能になる。

第 2 図及び第 3 図に例示したプロセッサ選択装

置において、レジスタ 80 を省略してゲート回路 70 の出力をマスタプロセッサの CPU が直接読取る構成とすることもできる。あるいはまた、プライオリティ・エンコーダ 40、レジスタ 31~3n の一方又は双方をマスタプロセッサの CPU からイネーブル、デセーブルする構成とすることにより、レジスタ 50、ゲート回路 21~2n 及び 70 を省略することもできる。また、プロセッサがタスク実行状態になつたことをマスタプロセッサの CPU から通知する例を示したが、この通知を各プロセッサから行う構成とすることもできる。

またプライオリティ・エンコーダ 40 の選択基準としてはシステム特性に合わせて適宜なものを選択できる。例えば空き状態を表示している複数台のプロセッサのうち最右端又は左端のものを選択する構成、あるいは無作為的に選択する構成、これらの選択基準を負荷状況に応じて変更できる構成とすることもできる。

以上詳細に説明したように、本発明は、各プロセッサが実行中のタスクを終了したときにこのプ

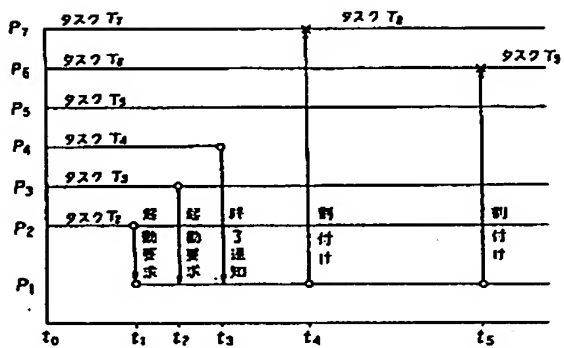
ロセッサが空き状態になつたことを表示するプロセッサ状態表示フラグをマスタプロセッサの CPU を介さずこのプロセッサから直接マスタプロセッサ内のレジスタにセットする構成であるから、CPU は空きプロセッサの存在を直ちに認識でき、わずかなハードウェアの追加によりシステム全体の処理能力を大幅向上できるという利点がある。

4 図面の簡単な説明

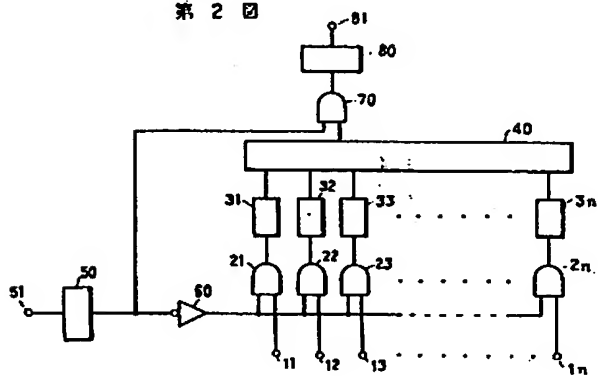
第 1 図は従来方式の問題点を説明するための概念図、第 2 図、第 3 図は本発明の一実施例に使用するプロセッサ選択装置の構成の一例を図示するブロック図である。

11~1n …入力端子、21~2n …ゲート回路、31~3n …プロセッサ状態表示フラグ・レジスタ、40 …プライオリティ・エンコーダ、50 …プロセッサ選択指示フラグ・レジスタ、70 …ゲート回路、80 …プロセッサ番号レジスタ、91~9n …入力端子、101~10n …動作可能表示フラグ・レジスタ、111~11n …ゲート回路。

第 1 図



第 2 図



第 3 図

